

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.



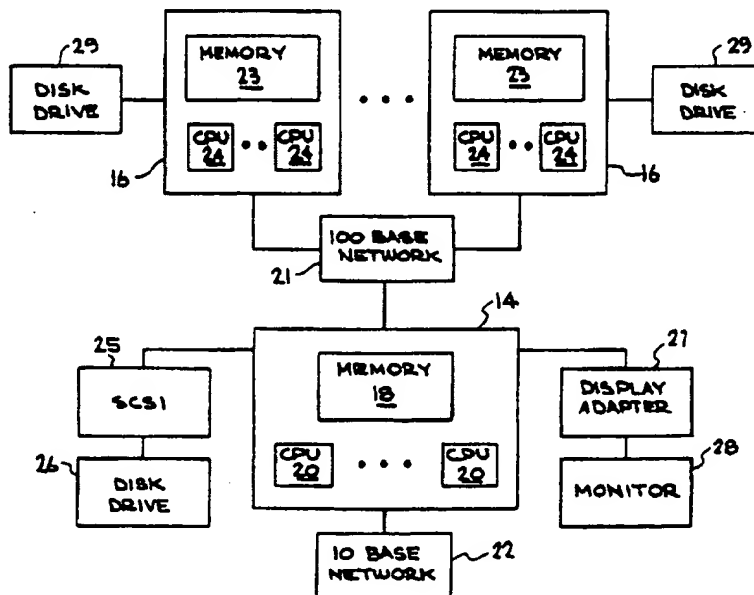
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G01T 1/161		A1	(11) International Publication Number: WO 97/42522
			(43) International Publication Date: 13 November 1997 (13.11.97)
(21) International Application Number: PCT/US97/07769		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 6 May 1997 (06.05.97)			
(30) Priority Data: 08/646,419 7 May 1996 (07.05.96) US			
(71) Applicant: THE REGENTS OF THE UNIVERSITY OF CALIFORNIA [US/US]; 22nd floor, 300 Lakeside Drive, Oakland, CA 94612 (US).			
(72) Inventors: MOSES, Edward, I.; 1075 Sherry Way, Livermore, CA 94550 (US). GARRETT, Dewey, N.; 2647 Wellingham Drive, Livermore, CA 94550 (US).			
(74) Agent: SARTORIO, Henry, P.; P.O. Box 808, Livermore, CA 94550 (US).			
		Published With international search report.	

(54) Title: RADIATION THERAPY DOSE CALCULATION ENGINE

(57) Abstract

As shown in the figure, the RTDC engine is a realizable design that implements the RTDC method for radiation treatment planning. The design makes practical the use of the superior performance and accuracy of a physics-based, Monte Carlo calculation to treatment planning systems used in working, clinical settings. The RTDC engine is based on available technology. It is made possible by exploiting the unique characteristics of the RTDC method. In particular, the engine maximizes the use of read-only, shared memory (18) for parallel, multithreaded compute processes to the limits of available bus architectures. Similarly, the design leverages the nature of the independence of the individual calculations to construct a hierarchical architecture that maximizes use of available resources for computation, storage and communication. Embodiments of the design can be constructed with emerging technology based on commodity computer equipment developed for server applications in areas of network and transaction processing. The architecture can be realized on a multiplicity of processor types and software operating system architectures.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

RADIATION THERAPY DOSE CALCULATION ENGINE

The United States Government has rights in this invention pursuant to Contract No. W-7405-ENG-48 between the United States Department of Energy and the University of California for the operation of Lawrence Livermore National Laboratory.

BACKGROUND OF THE INVENTION

Field of the Invention

5 The present invention relates to the use of radiation therapy to treat cancer patients, and more specifically, it relates to a calculation engine using a method for calculating the actual radiation therapy dose delivered to a patient, as disclosed in co-
10 pending patent application entitled "Use of All Particle Monte Carlo Transport for Radiation Therapy Dose Calculation" Serial Number 08/610,917, incorporated herein by reference. This method for
15 calculating radiation therapy dose, as disclosed in the above referenced patent application, is hereinafter referred to as the Radiation Therapy Dose Calculation (RTDC) method.

Description of Related Art

15 Currently in the United States, radiation therapy is used to treat about 60% of all cancer patients. Since radiation therapy targets specific areas of the body, improvement in radiation treatment techniques has the potential to reduce both mortality and morbidity in a large number of patients.

20 External beam radiation therapy is performed with several types of ionizing radiation. Approximately 80% of patients are treated with photons, ranging in maximum energy from 250 keV to 25 MeV. The balance are treated primarily with electrons with
25 energies from 4 to 25 MeV. In addition, there are several fast neutron and proton therapy facilities which have treated thousands of patients worldwide. Fast neutron therapy is performed with neutron energies up to 70 MeV, while proton therapy is performed

5,870,697

- 2 -

with proton energies ranging from about 50 to 250 MeV. Boron neutron capture therapy is conducted with thermal and epithermal neutron sources. Most internal radioactive sources irradiate the patient with photons, although some sources emit low energy electrons.

The effects of ionizing radiation on the body are quantified as radiation dose. Absorbed radiation dose is defined as the ratio of energy deposited to unit mass of tissue. Because tumors and sensitive structures are often located in close proximity, accuracy in the calculation of dose distributions is critically important. The goal of radiation therapy is to deliver a lethal dose to the tumor while maintaining an acceptable dose level in surrounding sensitive structures. This goal is achieved by computer-aided planning of the radiation treatments to be delivered. The treatment planning process consists of characterizing the individual patient's anatomy (most often, this is done using a computed tomography (CT) scan), determining the shape, intensity, and positioning of radiation sources, and calculating the distribution of absorbed radiation dose in the patient. Most current methods used to calculate dose in the body are based on dose measurements made in a water box. Heterogeneities such as bone and airways are treated in an approximate way or ignored altogether. Next to direct measurements, Monte Carlo transport is the most accurate method of determining dose distributions in heterogeneous media. In a Monte Carlo transport method, a computer is used to simulate the passage of particles through an object of interest.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a computation engine.

It is also an object of the present invention to provide a radiation dose calculation engine that uses the RTDC Method as disclosed in co-pending patent application entitled "Use of All Particle Monte Carlo Transport for Radiation Therapy Dose Calculation" Serial Number 08/610,917, for calculating the actual radiation therapy dose delivered to a patient.

- 3 -

The RTDC method for computing radiation dose in a patient volume relies on Monte Carlo methods that use proven physics models. It produces the most accurate results possible while providing statistical measures about the confidence of the calculation. To achieve this performance, the RTDC method must typically calculate millions of particle interaction histories. The results of these individual computations are incremental dose amounts that are distributed to the appropriate resolution elements (voxels) of the patient volume and summed to give the total dose at each voxel.

To deploy the RTDC method in a cost-effective manner, these computations must be made on affordable hardware with turn-around times that are useful in a working clinical environment. An apparatus deploying the RTDC method must support data interfaces that transfer data using proven, certified data structures and existing code libraries, and in addition, it must support physical interfaces to existing and future treatment planning systems.

In the present invention, a dose calculation engine, herein referred to as the Radiation Therapy Dose Calculation (RTDC) engine, addresses these needs through a flexible hardware and software architecture that is built from low-cost, commodity computer items that utilize modern operating system functionality. The RTDC engine architecture is designed to use state-of-the art components that are configured to maximize computational throughput in a scalable manner so that increased performance can be achieved by adding components. Moreover, the design provides capabilities for tuning and reconfiguration that allow it to overcome bottlenecks created by hardware limitations on bandwidth and it supports incorporation of new technology (as it becomes affordable) that will alleviate bandwidth constraints.

By nature, the RTDC engine is configurable in a number of embodiments that differ in cost, performance, and resolution. In addition, performance and resolution can be increased by adding components. These features enable offerings of the RTDC method and engine at a variety of market entry points that span a broad

- 4 -

regime of price, performance, and resolution. User investments are protected since upgrades that take advantage of cost reductions in components and memory are inherently supported by design.

5 The RTDC engine is built from hardware and software components that are currently becoming available and will be enhanced with future product development in several marketplaces. Specifically, the increased usage of the Internet and the ever increasing power of microprocessors are stimulating another transition of main-frame applications to new hardware. In the areas
10 of network servers and transaction-processing, suppliers are producing systems that are based on multi-cpu architectures running in a symmetric-multi-processor (smp) configuration. Typically such systems are based on a motherboard that contains 2 to 8 cpu microprocessor chips, memory, and connectors for peripherals and
15 disk i/o subsystems. In many cases the chips include a very fast internal instruction/data Level-1 cache and the logic needed to support large, fast Level-2 caches. Advanced chip designs include pipelined execution units, super-scalar architectures and advanced techniques such as speculative execution. These processor designs
20 directly accommodate smp implementations and incorporate features that address related issues of cache coherency and memory bus bandwidth.

 In the RTDC method, dose increments are computed by generating random particles (with appropriate statistics) and
25 propagating them through a patient volume described by computed tomography scans (CT) or similar data. As a particle propagates, dose increments are computed for summation into the volume elements (voxels) describing the net computed dose throughout the patient volume. A characteristic of the RTDC method is that it is especially
30 amenable to parallelization with low inter-process communications overhead. A common set of data can be used to describe the patient volume, the source data, and the problem setup. Multiple processes can share this memory in a read-only fashion. This read-only sharing minimizes issues of memory coherency between multiple
35 processors and simplifies locking and management of access to the describing data. For output, the RTDC method computes

- 5 -

independent dose increments that are localized to individual voxels of the patient volume. The total dose is computed by summing the individual dose increments. The independence of the separate calculations allows multiple processes to compute in parallel and send their results to a separate, independent process for summation. This separation of functionality enables a variety of mechanisms for management of the output sum. These mechanisms allow for buffering of dose increments and summation via immediate or deferred network communication (continuous or batch mode of dose update). The ability to buffer outputs results from the independence of the calculation and provides great flexibility in the construction, usage, and locking of the output dose memory.

Accompanying the development of high performance microprocessors are operating systems that incorporate symmetric multiprocessing and related functionality such as multi-threaded programming. These modern features allow performance to increase with the number of processors used. Depending on the nature of the problem and its data, computing performance of these machines can scale linearly with the number of processors used until a bandwidth limitation (bottleneck) is reached due to input/output requirements or memory usage. The RTDC method is inherently suitable to the scaling available with a symmetric-multi-processor (smp) design.

The availability of fast microprocessors, multi-cpu motherboards, and smp-capable operating systems makes a stand-alone RTDC engine possible. A huge marketplace and advancing technology continues to spur development of high performance microprocessors. Competition in the semiconductor, server, and software industries ensures continuing improvements in the performance/cost ratio. For example, suitable microprocessor designs include the following chip families:

	Intel	x86 (Pentium, Pentium Pro)
	IBM/Motorola	PowerPC
	DEC	Alpha
35	SUN	Sparc, Ultra-Sparc
	SGI	MIPS

- 6 -

HP

PA/RISC

In many cases, notably the x86 and Sparc families, high volume sales potential has created a competitive marketplace where additional vendors seek market share by offering compatible products that can run the operating systems and applications of the original hardware. Examples include companies like Cyrix, NexGen, and AMD (x86) and Ross (Sparc).

The availability of these microprocessor families has given additional impetus to the development of multi-threaded, smp-capable software operating systems. Current, suitable operating systems that support multiple hardware platforms include:

SUN	Solaris	(x86, PowerPC, Sparc)
Microsoft	NT	(x86, PowerPC, Alpha, MIPS)

These same companies are competing with compiler technologies and software development environments that support the development and testing of multi-threaded applications that scale in performance on multi-processor smp machines.

The use of these multi-processor machines for server architectures has also led to the development of low cost networking hardware and software for support of both 10 Mbits/second ethernet and 100 Mbits/sec fast ethernet at low cost. The breadth of this market arena ensures that higher speed communication solutions like FDDI, CDDI, and FibreChannel will be supported by the hardware applicable to a dose calculation engine.

The power and flexibility of many of these concepts has resulted in a strong, competitive marketplace for smp machines, smp-capable software architectures, and high speed networking technology. Existing and emerging standards address portable interfaces in areas of operating systems, networks, and multi-threading.

The RTDC engine design is built by taking advantage of the available high performance technologies, using standard interfaces, and maximizing performance by adapting its methods to current technology.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A shows RTDC equipment.

- 7 -

Figure 1B shows typical master-slave motherboard configuration within the RTDC equipment of Figure 1A.

Figure 2 shows the RTDC engine hardware in block form.

Figure 3 shows the RTDC engine hardware for a
5 continuous dose calculation in block form.

Figure 4 shows the RTDC engine hardware for a batched dose computation in block form.

DETAILED DESCRIPTION OF THE INVENTION

The RTDC engine is built with hardware based on multiple multi-cpu motherboards running the RTDC method on
10 software running on an smp-capable, multi-threaded operating system. Figure 1A shows RTDC equipment, including a master motherboard 2 and slave motherboards 4. The system of Figure 1A is shown connected to a local area network 6, which may be further connected to a treatment planning system. Figure 1B shows a typical
15 master-slave configuration within the equipment of Figure 1A. Master motherboard 2 is connected through internal network 10 to slave motherboards 4. Figure 2 shows the RTDC engine hardware in block form. The hardware is configured with a master motherboard 14 that handles interfaces, communications, and storage and a
20 multiplicity of slave motherboards 16 that run dose calculations in a parallel, scalable manner. The motherboards are identical for master and slave configurations but are populated with memory, CPUs, and peripherals appropriate to the allocation of tasks. Master motherboard 14 comprises memory 18 and a plurality of CPUs 20,
25 and is connected to a 100Base network 21 and a 10Base network 22. Slave motherboards 16 comprise memory 23 and multiple CPUs 24 and are connected to the master motherboard 14 through the 100Base network 21. The master motherboard 14 may be additionally connected to peripherals including a small computer
30 system interface 25, a disk drive 26, a display adaptor 27 and a monitor 28. Slave motherboards 16 may additionally be connected to a disk drive 29.

Each motherboard runs identical versions of an smp-capable operating system. The operating system configuration and
35 set of installed modules is tailored to the tasks assigned to the master

- 8 -

or slave motherboard. The master includes most classic operating system functions for interface, disk storage etc., while the slaves are configured to run with only the services needed to support calculation and communication with the master motherboard.

5 The basic hardware architecture is configured with a multi-threaded smp operating system support and can accommodate two distinct dose update configurations that allow the RTDC method to be tuned to improve performance for a variety of implementations. As product enhancements become available, this architecture is
10 adaptable in both its hardware and software configuration. This flexibility ensures that the product will support a long life-cycle that can adapt to user needs and accommodate both software and hardware capability enhancements.

 Motherboards developed for network servers and
15 transaction processing are suitable for both the master and slave functions of the RTDC engine. Typically such boards include a number of CPU chips and supporting logic circuitry, sockets for main memory, and connectors for the busses used to add expansion boards. Available motherboards may include additional circuitry
20 which is not required by the RTDC method.

 The RTDC engine uses identical motherboards for the master and slave configurations. However, the items installed on a given motherboard are different for the master and slave motherboards. The exact configuration of any motherboard can be
25 changed in order to add functionality or increase performance.

 The architecture can accommodate a variety of realizations; the primary required features are noted in the following descriptions.

 The RTDC engine uses motherboards designed for multi-
30 cpu, smp operation. A minimum of 2 cpu positions are required, 4 or more positions are a good match for the RTDC architecture (considering current memory and network performance), 8 or more positions are currently not practical but may become available in the future. The CPUs in all cases are used to run the smp-capable
35 operating system and tasks from the RTDC method. The CPUs on the master motherboard run tasks that handle interfaces, distribute

- 9 -

and collect data and programs, and provide real-time displays of the dose calculation. The CPUs on the slave motherboards run parallel user processes or threads that compute dose increments in a patient volume.

5 Motherboards with 8 or more positions currently include functions and complexities not needed by the RTDC engine or method and consequently must be individually evaluated for suitability. For maximum performance with a fixed number of motherboards, slave motherboards will populate all CPU sockets. A
10 master motherboard may use as few as 1 CPU or as many as the maximum number of slots depending upon the number of slave motherboards supported and ancillary functions assigned to the master.

15 RTDC engine motherboards have internal busses designed by the board manufacturer to facilitate communication between CPU's and memory. Because the boards are designed for smp operation, manufacturer's typically incorporate bus features to enhance memory-cpu bandwidth.

20 Additionally, the RTDC engine uses expansion busses to add peripheral functions required for operation. On a slave motherboard, expansion may be limited to a high speed network interface card which receives data and programs from a master and returns dose calculations to the master. The master motherboard uses a similar high speed network board or boards to communicate
25 with the slaves. The master motherboard also includes peripherals on the expansion bus for other functions. A typical configuration will include a disk i/o interface adaptor, a video display interface adaptor, and a network adaptor to interface with external hosts such as a treatment planning system host.

30 The memory requirements for the RTDC engine motherboards depends upon the dose update configuration and the target problem size. The RTDC engine uses motherboards that provide connectors to accommodate memory. With these designs, the amount of memory installed can be tailored to the application.
35 An RTDC engine master motherboard requires between 64 and 256 MBytes of memory; a slave motherboard requires between 32 and

- 10 -

256 MBytes. Additional memory may be added when required by future problem size increases or system requirements.

5 In all motherboards, memory is used to store executing modules of the operating system and RTDC method tasks. On slave motherboards, memory is also used to store data that describe the current problem (CT data, source data, nuclear data) and data representing the results of the dose calculation. On master motherboards, memory is used to store the supervisory tasks and the results of dose calculations accumulated from the slave
10 motherboards.

The master motherboard provides the hardware to support interfaces with external treatment planning systems plus peripherals for data storage, communication with slave motherboards, and an external video display.

15 The master motherboard minimum peripheral set includes disk input/output, internal and external network adaptors, and a video display adaptor. Additional peripherals can be added for functions (such as tape storage for backup) as required.

20 The master motherboard includes a general purpose input/output interface (for instance a fast/wide SCSI-II bus) that can support one or more disks. The disk is used to store the operating system and RTDC method programs and data. The disk can be used to store problem data and RTDC method results in order to facilitate transfers to a local host treatment planning system. A floppy disk or
25 other removable media device can be included to support interchange of data when a network connection is not available.

The master motherboard communicates with outside systems via a standard network connection. The specific network type can be chosen to match any common network; a TCP/IP 10Base
30 ethernet is a common implementation choice. This network is used to receive data and start-up instructions from a treatment planning host system. Data describing the patient volume (CT information), the source, and treatment information are transmitted using standardized file formats. The RTDC method application code is
35 designed to interface with AAPM and DICOM formats. Similar standards are used for return of the results from a dose calculation.

- 11 -

5 The external network is also used to receive high level instructions from the treatment planning system and its user interface. These instructions identify the data transferred, identify configuration parameters for the calculation, and initiate computation.

10 The use of a standardized external network allows the RTDC engine to support a number of standard interfaces to give it flexibility in both local and distant communications. The RTDC engine will optionally provide files services (both exporting and importing) using the industry standard Network File System (NFS). Other standard interfaces like the X Window system, will be available to monitor and configure the RTDC engine.

15 The Master motherboard communicates programs and data to slave motherboards via an independent, high speed network interface. The network is operated independently from the external or host network so that the RTDC engine's internal, intermediate data transfers are not affected by outside activity. This network is used to distribute programs, data, and instructions to the slave motherboards and receives dose calculation results from the slave motherboards. The potential for high transfer rates mandates a high speed network; 100 Mbit/sec fast ethernet is a suitable, low-cost interface for this network. The dose calculation results may be received by the master motherboard in different forms depending upon the installed dose update configuration (continuous or batch).

25 In the continuous dose update configuration, as shown in Figure 3, the internal network 30 receives dose increments continuously from the slave motherboards 32 as their CPUs 34 perform the Monte Carlo calculations. Each slave motherboard calculation result is sent to a tunable buffer 36 and ultimately sent to the master motherboard (MMB) 38. The MMB 38 receives the result into collector 40, from which a code 41 performs statistical analysis at 42. Each data transfer includes an index describing the appropriate voxel to receive the increment. Collector 40 sums the dose at 42 which is interfaced to an external network 43, a video adaptor 44 and a monitor 45. Each motherboard has an operating system 46
35 comprising a multi-threaded smp capable operating system. This

- 12 -

method greatly reduces requirements on slave memory at the expense of bandwidth usage on the internal network.

In the batch dose update configuration, as shown in Figure 4, the CPUs 50 on the slave motherboard 52 compute dose and accumulate directly into a local memory 54. At intervals, the entire dose accumulated by a slave 52 is transferred through internal network 56 to a collector 58 on the master motherboard 60 which sums the received dose volume to an accumulation volume 62. This method alleviates bus bandwidth limitations at the expense of the local memory required by each slave. The motherboards all include code 64 for performing statistical analysis, such as variance, which is summed into 66. Each motherboard also includes a multi-threaded smp capable operating system 68. The bandwidth required is substantially less than in the continuous update configuration since individual indexing of dose increments is not required and the update rate is not crucial with respect to attaining overall performance. The master motherboard includes an external interface 70, a video adaptor 72 and a monitor 74.

A unique feature of the RTDC engine is the capability of showing the accumulation of dose and statistics about the calculation on a real-time display. This display is made possible in both the continuous and the batch dose update configurations since the master motherboard contains the final, dose accumulation memory. A simple process on the master motherboard is used to read the dose and send it to a display via a video display adaptor peripheral.

The real-time display of computed dose will give the operators and clinicians an unprecedented capability to view the results of a treatment plan computation. The display will show the computational progress and indicate with statistical measures the increasing confidence of the calculation. The operator will be able to accept results with bounded confidence levels and can extend computation time to improve the statistical confidence of a treatment plan. This display will rapidly indicate the effects of multiple beams to allow successive iteration on a plan. Errors in

- 13 -

plan specification will appear rapidly so that mal-formed plans can be aborted and redescribed.

5 The RTDC engine can accommodate one or more slave motherboards to implement dose calculations. Each slave motherboard includes the common items (CPUs, memory, and peripheral busses). The number of CPU's and the amount of memory may be increased to improve timing performance or problem size capability respectively.

10 Slave motherboards can be configured to initialize (boot) from network services provided by the master motherboard. In this configuration, transfer of system modules, the RTDC method applications code and its data is accomplished via the internal network and no supplementary disk storage is required. To support this configuration, the only peripheral needed is an internal high speed network adaptor.

15 Additional peripherals such as a local disk or a video display can be added to a slave motherboard to facilitate setup or testing. A small local disk can be used for boot services for configurations which do not support remote booting. The disk image need contain only operating system support since additional resources can be obtained by the Network File System operated over the internal network.

20 Slave motherboards receive programs and data across a high speed internal network. Slave motherboard dose calculation results are transferred back to the master motherboard in either the continuous or the batch update method described earlier.

25 The RTDC engine implements the RTDC method applications software in a distributed multi-machine, multi-cpu configuration. This configuration takes advantage of the characteristics of the RTDC method and utilizes available commodity, server-class computer equipment and modern smp software to implement a low-cost system.

30 The RTDC method applications code is implemented in a context provided by the operating system software and utilizes readily available and portable support software to implement local interfaces and the distribution of parallel tasks. The RTDC method

35

- 14 -

itself can be configured in different update configurations (continuous or batch) to take advantage of available memory and to reduce the affects of network bandwidth limitations.

Several characteristics of the RTDC method make it particularly suitable for modern symmetric-multi-processor (smp) operating systems and multi-cpu machines. In particular, multiple processes can share in a read-only manner the memory that describes the current problem -- the patient CT scan information, the source and plan description, and nuclear data. For output, the same parallel processes can send computed dose increments to a separate, independent process that handles summation of increments at each volume element. This independent summation facilitates buffering of dose increments for efficiency and minimizes issues of contention and lock management on the dose memory. Furthermore, the independent summation supports a variety of dose update methods that allow for performance scaling by addition of processors and supports batched solutions to address ultimate network bandwidth limitations. The output process is organized in a simple hierarchical fashion (master-slaves) and can be extended to more levels of hierarchy for future problems involving huge datasets or extreme history accumulation requirements.

The usual benefits of smp processing are used to advantage in the RTDC engine. In particular, the use of multi-threaded techniques allows a single software design to be utilized on slave motherboards having any number of processors. The efficient, lightweight methods available with thread programming make possible a high degree of tuning in terms of the number of calculating threads and their output buffer sizes. This tuning allows an n processor machine to have m (not necessarily equal to n) calculating threads as well as a collection thread and ancillary operating system processes. By appropriate design and tuning, all system resources (memory, disk, and network i/o) are utilized to near maximum capacity.

Because of the features outlined above, the RTDC requires an smp-capable, operating system with multi-thread support. This requirement is readily fulfilled by most modern UNIX

- 15 -

implementations and by Microsoft's NT. These implementations additionally provide necessary libraries for utilities, input/output, and mathematics functions simplifying the portability of the RTDC method. The availability of multiple operating system choices
5 ensures that the RTDC method implementation can be maintained with modern, available operating systems on commodity, server hardware.

Where possible, the RTDC engine takes advantage of de facto and published industry standards. In particular, availability of
10 the ubiquitous Network File System (NFS) and the X Window System (X) help to standardize external interfaces to treatment planning systems. Use of these standards ensures that the RTDC engine can interface to most architectures used in present and future treatment planning systems. In developing areas such as thread
15 application libraries, the RTDC engine can use proprietary (e.g., operating-system-specific) or standards (Posix pthreads) as they are developed and supported by vendors.

The master motherboard is fully configured with most standard operating system functionality in order to support a
20 treatment planning host and a full range of services for the RTDC method. It includes sufficient networking services to communicate with external name servers, routers, and hosts on the external network and can provide boot services to slave motherboards on an internal high speed network. The master motherboard includes
25 support for disk subsystems and graphics display adaptors used for storage and real-time display of the dose calculation results.

The slave motherboards can be operated with a minimum set of operating system modules and services in order to reduce overhead for resources not needed for Monte Carlo calculations.
30 Allowing slave motherboards to boot via the internal network from the master motherboard can eliminate the need for a local disk and simplify systems update and maintenance.

In summary, the RTDC engine is designed to be compatible with a variety of modern smp-capable, multi-threaded operating
35 systems. For the type of low-cost, server-based hardware needed,

- 16 -

several available operating systems exist ensuring the present and future viability of the RTDC engine.

In addition to the RTDC method and the operating system software, the RTDC engine requires supporting software for
5 distribution of tasks between master and slaves and for implementing external interfaces for download of source, testing, and configuration.

Interfaces to an external treatment planning host computer are largely handled by industry standard utilities and services that
10 include UNIX or "UNIX-like" shells and the Network File System (NFS). Because of the complexity of the RTDC engine hardware and software configuration, additional interfaces are needed to support local setup, configuration, and test. Related to these requirements is the need to provide a real-time display of the dose calculation with
15 user adjustment of selected display parameters. In order to fulfill these requirements, the RTDC engine relies on a set of tools that are well-tested, robust, and portable. A preferred toolkit named Tcl/Tk, originally developed by researchers at UC Berkeley, provides a
20 simple scripting language (Tcl) and graphics interface (Tk) suitable for these tasks. These tools are portable between all UNIX systems and are being ported to NT and Windows95. Availability of these ports provides access to the test and configuration interfaces of the RTDC engine by low-cost and readily available systems attached to the RTDC engine's external network.

25 The scripting languages are used to setup and tune the RTDC engine. Tuning parameters include the number of processes executing on slave processors and the size of buffers used for collection of dose output increments on both slave and master motherboards.

30 The scripting language in conjunction with a graphics toolkit is used to provide local viewing of the computed dose in real time. This capability is useful in testing operation of the RTDC method and give a clinical user instantaneous feedback about the suitability of the plan, progress of the computations, and final results
35 of the computations. The local display allows the user to select a portion (e.g., a slice) of the patient volume and display the results of

- 17 -

the dose calculation with a set of colors chosen to illuminate user-selected features. The availability of real-time dose calculation display is expected to lead to new capabilities in treatment planning by allowing the clinician to reject, modify, or add to an original treatment plan.

The choice of an existing toolkit language with scripting capabilities allows providers of the RTDC engine to provide rapid development of interface and test functions that add value to the application. This environment is implemented on a local level in order to supply test and tuning features and in addition provides a means to demonstrate capabilities that can be evaluated for future inclusion in the dose engine. This means of presentation is offered in addition to the proven and certified functionality so that it does not interfere with validated requirements. With this approach, providers can illuminate the potential use of new viewpoints and methods made possible by the RTDC method and the display of calculation results in real time.

The RTDC method is an ideal candidate for parallel computation since it can effectively use parallel processes that 1) can be multi-threaded, 2) can share the same describing datasets in a read-only manner and 3) can compute and transfer dose calculations independently. These capabilities in conjunction with symmetric multi-processing make possible the low-cost hierarchical design of the RTDC engine.

Because of these capabilities, distribution of tasks and communication between a master and its slaves can be accomplished with a low overhead costs for interprocess communication. The principle tasks involve the initialization of data structures and the start up of threads on the slave motherboards. All slave processes (amongst all slave motherboards), communicate only with coordinating and data collection threads after startup. The lack of inter-process communication between the computing processes and the dominance of uni-directional data transfer greatly simplifies interfaces and messages between processes.

With these considerations, the RTDC method is able to take advantage of existing, robust implementations for distribution

- 18 -

of parallel tasks. Several methods have been developed by numerous researchers for this type of distribution. The RTDC method is well suited to the methods of the Parallel Virtual Machine (PVM) originally developed by the researchers at the Oakridge National Laboratory (ORNL) and the University of Tennessee, Knoxville (UTK). This implementation has been ported to all standard UNIX operating systems, has well documented interfaces to common languages including those (C, Fortran) used by the RTDC method, and include a script/graphical interface (tkPVM) that is implemented with the Tcl/Tk toolkit.

For the RTDC method, PVM is used in a simple form to distribute the RTDC method computations code and data to slave Motherboards. Additional distribution of tasks that measure and profile performance and provide developers the information needed to further improve code are provided through standard PVM resources.

The suitability of PVM to the RTDC engine reduces the development time and cost to field initial and upgraded versions of the RTDC engine. Moreover, built-in capabilities of PVM for message handling functionality and the support of heterogeneous systems provide for adapting the RTDC method to new areas with different requirements for interprocess communication and diverse operating environments.

Since its inception, the computer industry has been characterized by increasing speed and performance at declining cost. At any given time, hardware and software designers must adapt their designs to suit the availability of components and their limiting characteristics. In general, the final limit on speed-cost-performance (however characterized) is termed a "bottleneck" and designs are modified to alleviate its affect. The design of a cost-effective, Monte Carlo-based dose calculation engine has been made possible by the rapid development of computers, operating systems, and network technologies. Nonetheless, the RTDC engine must provide solutions for "bottleneck" limitations to provide a viable, long-lived vehicle in the competitive field of treatment planning.

- 19 -

The RTDC engine design can readily incorporate increases in processor speed and performance and advances in software design. Moreover, the calculation engine is inherently scalable by adding processors to slave motherboards and by adding entire slave motherboards. If this process is continued, however, ultimately a limit is imposed by the bandwidth of the high speed internal network that funnels dose increment calculations for summation. Solutions for this bottleneck category are available in several ways including alternate methods for dose update configuration designed into the architecture.

At the time of its design and for the class of problems envisioned, the RTDC engine uses a continuous dose update configuration (see below). This method uses low amounts of total memory while providing continuous update of dose calculations throughout the patient volume. The design can be functional with a single slave motherboard populated with a single CPU and becomes viable commercially when a total of 8 to 12 CPUs are configured via slave motherboards with smp-architectures. As scaling is increased to more than 16 CPUs, the bandwidth of the internal network is expected to become the limiting bottleneck. Solutions for this limit may include 1) incorporation of higher speed internal networks or busses, or 2) replication of networks with a dedicated network for each slave. The first solution is dependent on the availability of low cost hardware, the second is viable but in all likelihood moves the bottleneck from the network to the master motherboard internal bus.

The RTDC engine architecture offers another method to reduce the effect of internal network data saturation that is implemented at the expense of additional memory. This batch dose update configuration method (see below) is an attractive solution because it is planned within the architecture and is accomplished by trading off performance with memory and cost.

The most cost effective implementation for the RTDC engine on smp-machines is the continuous dose update method. In this configuration, each slave motherboard uses a single local copy of the problem describing data sets that are used read-only by all

- 20 -

processes running on all slave CPUs. The computed outputs of each process are dose increments that are localized to a set of three dimensional coordinates in the patient volume. These results are buffered locally and transmitted to the master motherboard for final accumulation of all results. This design requires relatively low amounts of memory in the slave processor since the outputs of the calculation are just buffered and sent to the master. The reduction of memory is especially important whenever memory costs are high, problem sizes are large, or many slave motherboards are used in an implementation.

The continuous dose update method supports a cost-effective distribution of work that allocates dose summation to the master motherboard. An additional benefit of this allocation is the single location and near real-time validity of the calculated result over the entire patient volume. This configuration lets the master motherboard provide real time displays of dose in a nearly continuous manner and supports real time measurement of the statistics and performance of an ongoing computation.

In the batch dose update configuration, each process on a slave motherboard computes dose increments and coordinants identically to that described for the continuous dose update configuration. For batch update, however, the computed dose increment is summed directly to a local dose accumulation memory. This array is transferred entirely to the master at intervals. Since the transfer sends the entire dataset at a time, specification of individual coordinants is eliminated. These characteristics make the transfer highly efficient in the use of the internal network. Simple compression schemes, for example, run-length-encoding, may be used to further increase efficiency.

The main tradeoffs of the batch dose update configuration are the increased use of memory on slave motherboards and the additional coordination required by the master motherboard to gather and sum entire dose arrays. A side effect of this implementation is the increased time granularity of the displays of total dose performance or statistics measurements.

- 21 -

The detailed tasks and task distribution are largely the same for this method as for the continuous dose update method. Since they are so similar, the RTDC engine design can incorporate both functions as compile-time options. Overall, the availability of the two methods provides ongoing ways to develop and improve code that can be tested on different variations of the same architecture to ensure the flexibility needed when market conditions and available technology change.

Operation of the RTDC engine is performed in a number of steps involving the master and slave motherboards. The primary functions involved are 1) initialization, 2) problem distribution, 3) dose calculation, and 4) dose collection. Details of the dose collection function differ depending upon the implementation of the dose update method (continuous or batch). In addition to these primary functions, the architecture supports additional capabilities that can be performed on processes of the master motherboard. These ancillary functions can include a continuous display of the dose calculation and adaptive control which uses intermediate results or user input to provide feedback to alter the dose computing processes behavior.

Descriptions of the primary and ancillary functions follow.

At start-up the master motherboard performs routine system checks and establishes and verifies its connection with a treatment planning system host. It also verifies the connectivity of attached slave motherboards on the internal high speed network and provides boot services to transfer the operating system and the loadable modules required for slave operation. Each slave motherboard boots from the internal network (or a local disk if one is installed) and establishes a connection with the master motherboard for assignment of RTDC method tasks.

After the initialization steps, the RTDC engine is ready to receive plans and data from a treatment planning system host. The interface that the RTDC engine provides is flexible so that this information can be transferred in a variety of ways. The principle methods can be summarized as interactive and batch. (Here "batch" is used to connote delivery of a queue of jobs which are executed on

- 22 -

a first-in-first-out basis without operator attention.) Within these broad categories, the RTDC engine can receive data by copying files or by using standard network services such as the Network File System (NFS).

5 For interactive job submittal, data describing the CT scans, machine configuration, and treatment plan are received in standard formats (AAPM or DICOM). For batch job submittal, a list of separate cases are prepared and sent to the RTDC engine to establish a queue of jobs. For this mode, jobs are executed sequentially from
10 the queue without requiring operator attention.

 For the case of interactive job submittal, the RTDC engine provides a mechanism that supports early acceptance of data in order to reduce the "apparent" time for problem computation. This feature allows the treatment planning system to specify and transfer
15 voluminous CT scan information as soon as it is identified in the treatment planning process. The RTDC engine can receive this information and distribute it to the memory units of the slave motherboards while the operator completes the planning process. This anticipatory distribution of data eliminates a significant time
20 delay at problem start-up and improves the "apparent" time of problem solution in interactive mode.

 When all components of a job are received, the RTDC engine master motherboard verifies the self-consistency of the describing files and initiates problem distribution to the slaves. This
25 step includes transmission of CT data (if not already sent), data describing the particle source machine and its components, and data describing the plan. The data is derived from the standardized files and formats used (AAPM, DICOM) but may be formulated as a memory image for immediate use by the calculation routines of the
30 slave motherboards.

 The RTDC engine can distribute large data sets from the master to the slaves by a variety of means that reduce the time delays incurred. In many cases, the problem describing data is identical for all slaves so the master motherboard can make effective use of
35 broadcast or multicast protocols on the internal high speed network.

- 23 -

This technique distributes to all slave motherboards simultaneously eliminating unnecessary repetition of data transmissions.

Once the data describing the problem and plan are distributed, the master motherboard can initiate computation on
5 processes on slave motherboards. Initiation of tasks is handled through standard, robust utilities such as the Parallel Virtual Machine (PVM). The startup of individual compute tasks or threads is parameterized so that a single, master supervisory process can provide the means to tune problem solutions and provide iteration
10 and feedback methods to an implementation.

The RTDC engine architecture supports variations in the distribution techniques in order to adapt to specific hardware implementations. Tunable parameters include the number of
15 computing threads initiated, work allocation strategies, and iterative computation. These parameters are available within either the continuous or batch update configurations.

Varying the number of computing threads is a simple but effective method to maintain calculation rates when a computation thread must wait for resources. For example, when a computing
20 thread fills its buffer of dose increments, it must wait for a collecting thread to consume the buffer before it can continue computations. By creating additional processes using simple multi-threading techniques, computation can continue transparently on all available CPUs when a given thread is blocked for output. Alternatively, a
25 blocked compute thread may acquire an empty buffer from a managed pool of buffers and resume computations immediately.

The RTDC engine can distribute work in a variety of ways to improve visibility of ongoing results or to take advantage of hardware design features. Work distribution can include simple
30 allocations of specific beams to individual processes or assignment of particle statistics to different slave motherboards or processes.

The master motherboard supervisory processes can coordinate iterative computations that enhance the operator's view of the computations results or support feedback in the process. In
35 the batch dose update configuration, iteration is a necessary feature that is used to coordinate batch updates of the final dose summation

- 24 -

memory. The iterative method has additional advantages in that it allows slave processes to compute with smaller data sizes since long data structures are only needed for the final dose summation. This division of data structures helps to avoid errors inherent in summing large quantities of small values while distributing memory in a cost effective manner. In the continuous update configuration, iterative methods are not mandatory but provide a means for feedback control from the master to the computing processes.

In addition to the initiation of compute threads, the master motherboard creates and starts processes which collect data from the compute threads and transmit it for final summation on the master motherboard. While the implementation details of these collection/transmission tasks is highly dependent on the dose update configuration (continuous or batch), the startup mechanisms and final summations are similar for all implementations.

Processes or threads running on the CPUs of the slave motherboards implement the RTDC method algorithms for dose computations. The computer code that implements these algorithms is highly optimized for rapid, accurate modeling of the appropriate physics. The use of smp facilities allows a multiplicity of these processes to be started and executed simultaneously on each slave motherboards.

The RTDC method reads data describing the CT scans, the machine or source characteristics, and the treatment plan information from a common memory accessed in read-only mode. Usage of memory in this manner is efficient and makes good use of the memory caches associated with modern microprocessors. Results of the computation are characterized as "dose increments". These results represent an amount of energy at a specific volume element (voxel) within the patient volume. Each result must be summed in a data structure representing all voxels of the volume.

All computations of dose increments are independent so the ordering and timeliness of the final summation is not a crucial consideration with respect to a compute thread's functionality. The RTDC engine is able to take advantage of this important

- 25 -

characteristic by allowing the computing processes to buffer results in a manner that makes the summation efficient. In a typical implementation, a computing thread will be allocated a buffer to receive computation results. The thread can compute dose
5 increments, verify buffer space availability, deposit its results in the buffer, and immediately continue its computational tasks. Data is taken from the buffer by independent, collection processes specialized for the chosen dose update configuration (continuous or batch). This design follows classic methods in computer I/O where
10 input/output buffering is used to provide efficient interactions with producers and consumers of data. In the RTDC engine compute thread case, the use of an output buffer minimizes the time the thread spends on any non-compute work (a single test for buffer saturation suffices for the compute task). The buffer size is tunable
15 to take advantage of details of the implementation such as the number of CPUs, processes, available memory, and network speed and latency.

Dose collection and summation are performed by a number of cooperating threads or processes on the master and slave
20 motherboards. The implementation is structured according to the selected dose update configuration.

In the continuous dose update configuration, slave compute threads deposit calculated dose increments into thread-specific buffers of tunable size. The dose data structure includes a
25 dose increment amount and an index that locates a specific volume element in the patient volume. The buffers from all threads running on a slave motherboard are read by a dose collection thread. This thread reads a buffer at intervals and forwards the collected data to the master motherboard through transmissions on the high speed
30 internal network. On the master motherboard, a complementary dose collection thread receives data transmissions from all slave motherboards. It performs the summation of dose increments to a single, final dose accumulation memory. An important characteristic of this method is the minimal amount of overhead
35 associated with communications. The slave compute threads simply write to a buffer, the slave collection thread reads and transmits

- 26 -

5 buffers, and the master collection thread reads transmissions and sums increments into memory. No complex locking or multiple write-accesses to data structures are required and all interprocess communication is handled through buffers that are tunable in size. This method is highly efficient in memory usage since only a single, volume-sized dose accumulation memory is required. This same methodology allows compute threads to use smaller data structures for computation since the size of roundoff errors due to the summation of small numbers is managed by use of large data sizes on the final accumulation memory.

10 In the batch update dose configuration, slave compute threads behave in the same manner as for continuous update. The slave collection thread, however, now sums directly to a local dose accumulation memory. An additional slave transmission thread periodically transmits the entire local accumulation memory on the high speed internal network to a complementary process on the master motherboard. The use of buffers between the compute threads and the collection thread facilitates transmission of the entire local accumulation memory with minor interruptions to the compute threads' continuity of execution. This method trades increased memory requirements on the slave motherboards for reduction in requirements on internal network bandwidth. The master motherboard thread responsible for collection and summation of dose is now configured to receive and sum entire volume arrays for summation to a final dose accumulation memory. Overall, this method retains key operating characteristics that support efficient communication. Its use of buffered output from compute threads and single writers to both local and master accumulation memories maintain the simple, contentionless mechanisms used throughout the design. In addition, the batch design moderates the amount of slave motherboard memory required by limiting the batch size so that relatively small data structures can be used. This strategy allows summation caused roundoff error to be managed via control of the batch "size" (quantity of local summations) and the data structure size used in the master final dose accumulation memory.

- 27 -

In all dose update configurations, supervisory processes on the master motherboard are able to monitor progress so that the computations can be stopped after solutions are attained with required, predescribed statistical qualifications. These same processes
5 can respond to input from the supervisory treatment planning host to abort running jobs or reschedule items in a job batch queue.

At completion of the dose calculation in interactive mode, results and job log information are transferred back to the treatment planning system host. Once the transfer is initiated, the RTDC
10 engine is available to respond to subsequent interactive job submittals.

When running in a batch job submittal mode, as soon as the concluding data transfers are started, startup of a new job is begun to repeat the sequence of problem distribution, dose collection
15 and summation.

The power and flexibility of the architecture of the RTDC engine support additional functions that will increase its value in a clinical setting and create new opportunities for treatment planning system vendors. This flexibility is made possible by the
20 architecture's repeated use of a single, general-purpose, server-based motherboard for both master and slave functions. Typically, a motherboard will have 4 to 8 CPU positions available and slave motherboards will be fully configured with CPUs. The master motherboard which performs interface and dose accumulation tasks
25 is configured to run on the same type of motherboard with fewer CPUs in a configuration that supports more peripherals and memory. The master motherboard typically handles interface, problem distribution, and data collection/summation tasks. In the batch dose update configuration, many of these functions operate in
30 a burst manner. In the continuous update configuration, data collection is more intensive with work focused on network activity. Both configurations are amenable to the incorporation of additional work by using idle processor time or additional CPUs. The availability of non-CPU resources between bursts of activity or while
35 blocks occur due to network buffer filling can be utilized for additional tasks. The availability of extra CPUs and symmetric

- 28 -

multiprocessing support ensures that ancillary tasks can be added with minor consequences to the engine's dose calculation throughput.

5 The initial ancillary tasks envisioned relate to the generation of a real-time display of dose computation results and incorporation of feedback methods to improve calculation performance or its visualization. The availability of this type of display is expected to lead to new functionality that can incorporate user feedback into the treatment planning process. Future ancillary
10 functions can include implementation of both new treatment planning functionality and tasks which offload or assume the functions of a treatment planning system host.

15 The master motherboard can spawn a thread or process that concurrently reads the dose computation result as it is summed into the final dose accumulation memory. Because the master motherboard includes all facilities of a general purpose computer, commonly available functions can be adapted to create displays of the dose on an attached video display peripheral. The display can be organized to show any two-dimensional section of the three-
20 dimensional volume memory that accumulates dose. In addition, displays of related reference information, namely the CT scan of the patient can be overlaid or otherwise incorporated to give context to the dose display.

25 Typical 2D displays would show a slice in a plane related to one or more injected beams of the plan. Enhanced displays can show views and animations of the projection of three dimensional volume-renderings of the dose and the reference CT scan information. The video display can also be equipped to allow the user to alter the display in order to select a particular slice or volume
30 region and to apply pseudo-color mapping schemes that highlight behavior in ways that are both intuitive and meaningful.

35 The display of the computed dose in near real-time gives the operator an instantaneous view of the treatment plan's development. Because the RTDC method computational algorithms model the actual physics of the treatment process, the real-time dose display closely follows the actual physical processes. Initially, the

- 29 -

dose computed in the volume appears as a noisy, random spatial accumulation but it quickly consolidates as dose builds up with time. The accumulation of dose in absorbing tissue and the non-accumulation in air and air cavities is quickly illuminated. The effects of multiple beams is clearly shown; the rapid and focused build-up at the intersection of beams is dramatic and useful in showing a plan's effectiveness. To experienced radiologists, the display reinforces and confirms the details of the plan; to trainees and observers the display can give new meaning to the abstract and complex physics used in the methods of treatment. The display effectively shows how the RTDC method computation models the physical processes of radiation treatment and leads to new insight into methods of treatment planning.

The display shows in an integrated way the effect of each beam and the dose accumulation at the intersection of multiple beams in the three dimensional patient volume. Availability of the reference patient CT scan displays allows a rapid assessment of the effectiveness of the plan with respect to tumor targets and protection of nearby vital tissue. An obvious immediate benefit of the display is that erroneous or malformed plans can be recognized and abandoned without requiring computation of a complete planning solution. In the future, it is expected that the user will be able to improve plans based upon the intermediate results presented -- the RTDC engine architecture is well-positioned to accommodate iterations to an initial plan.

The RTDC method algorithms are based on statistical methods that give increasing confidence in the calculation result with increasing time. The increase in confidence is manifest in the real-time dose display as the initial random or noisy accumulation becomes more and more consolidated as the calculation proceeds. The increasing consolidation that is visible to any human observer is directly related to the improving statistics and confidence of the calculation. This information will be used by operators in significant ways. After gaining experience and understanding the real-time dose display, operators will be able to adjust their calculation criteria in ways that improve the efficiency of the entire treatment planning

- 30 -

system. Simple treatment plans for cases that are not constrained by issues for protection of adjacent sensitive tissue will be handled with routine, built-in settings for computational performance at a predescribed statistical confidence level. Cases with complex requirements, including concerns for adjacent tissue, can be run at the operator's digression for additional time with resulting increased statistical confidence in the computed result.

The RTDC engine architecture uses a master-slaves approach that allows low cost hardware to be used effectively for parallel computation of dose. A characteristic of the design is that the final dose accumulation memory is located on the master motherboard which can perform additional calculations on the collected result without slowing down the primary work executed on the slave motherboards of the machine. In addition to displaying the computed dose, master motherboard processes can be created to perform new functions that add value or improve overall system performance. The architecture is readily amenable to the addition of processes that perform additional calculations based on the current dose computation result and its progress. These additional calculations can be used in a variety of ways to 1) quantify the current statistical confidence of the calculation, 2) stop calculations after prescribed confidence levels have been attained, 3) detect hazards and abnormal situations created by an operator or malformed input, 4) alter (via feedback to the computing threads) the problem description in a way to improve performance globally or within specific local volumes.

Implementation of adaptive behavior is especially suitable in the continuous dose update configuration. In this configuration, the master motherboard receives all dose increments via the high speed internal network and has the principle task of summing increments into the dose accumulation memory. Additional processes can be added to perform additional calculations on incoming dose increments on an individual or sub-sampled basis. The addition of new computations for each incoming dose increment is obviously expensive but can be accommodated on the master motherboard by adding additional CPUs and taking

- 31 -

5 advantage of the architectural and smp features available throughout the design. The alternate method of sampling dose increments is a more effective method that can provide valid measures on progress and statistical confidence with moderate use of computer resources. Implementation of adaptive behavior in the context of the batch update configuration can use all the methods available for the continuous dose update configuration but requires distribution of parallel processes to the slave motherboards in many instances. In addition to the increased complexity, most methods will require additional memory that must be replicated with the new processes on the slave motherboards. The tradeoffs associated with ancillary adaptive computations are similar to those for dose computation in general -- computations made in batch mode will require more total memory but not add greatly to bandwidth limited data flows. Continuous mode ancillary computations can use memory efficiently but exacerbate the aggregate amount of data flow at some point. The situation is greatly aided for ancillary computations that can provide information when performed on a sub-sampled basis.

20 An obvious, useful ancillary computation is the statistical variance computed for each dose volume element. While expensive to compute, the variance gives a detailed measure of the progress of the computation throughout the entire volume. This information can be used to perform additional analyses that aggregate volumes with common materials and volume element statistics in a way that supports additional measures on the confidence of the calculation throughout the volume. Less expensive computations can be made by identifying smaller, important tissue volumes that represent either tumor or critical tissue. The results of either type of analysis can be used in a number of ways that include 1) terminating the calculation when confidence criteria are achieved, 2) altering the problem description by changing the resolution basis in certain volumes, 3) altering the details of the Monte Carlo computation to exploit well-known variance reduction techniques on specified volumes, 4) consolidation of computed dose

- 32 -

calculations into smoothed displays that combine small, adjacent volumes that have similar characteristics.

5 Changes and modifications in the specifically described embodiments can be carried out without departing from the scope of the invention, which is intended to be limited by the scope of the appended claims.

THE INVENTION CLAIMED IS

1. A radiation therapy dose calculation (RTDC) engine, comprising:

a master motherboard comprising memory, a plurality of CPUs, peripherals, interface means and communication means;

5 a plurality of slave motherboards comprising memory, a second plurality of CPUs, peripherals, interface means and communication means, wherein each slave motherboard of said plurality of slave motherboards is controlled by and in communication with said master motherboard, wherein said
10 plurality of slave motherboards comprise means for performing parallel dose calculations comprising a computer implemented process for producing a 3-dimensional map of a radiation dose delivered to a patient;

a multi-threaded, symmetric-multi-processor capable
15 operating system running on said plurality of CPUs and said second plurality of CPUs, wherein said operating system provides services and coordinates concurrent execution of said parallel dose calculations.

2. The RTDC engine of claim 1, wherein said computer implemented process for producing a 3-dimensional map of a radiation dose delivered to a patient, comprises:

5 constructing patient-dependent information necessary for a Monte-Carlo transport calculation;

executing said Monte-Carlo transport calculation; and

producing, from said patient-dependent information and said Monte-Carlo transport calculation, a 3-dimensional map of the dose delivered to said patient.

- 34 -

3. A calculation engine, comprising:

5 a master motherboard (MMB) comprising memory and a plurality of CPUs, wherein said MMB is connected through a peripheral interface to a disk drive, wherein said MMB is further connected to an internal Network and an external Network;

10 a plurality of slave motherboards, wherein each slave motherboard (SMB) of said plurality of slave motherboards comprises memory and a second plurality of CPUs, wherein each said SMB is connected to a disc drive and wherein each said SMB is further connected through said internal Network to said MMB, wherein said plurality of slave motherboards comprise means for performing parallel calculations; and

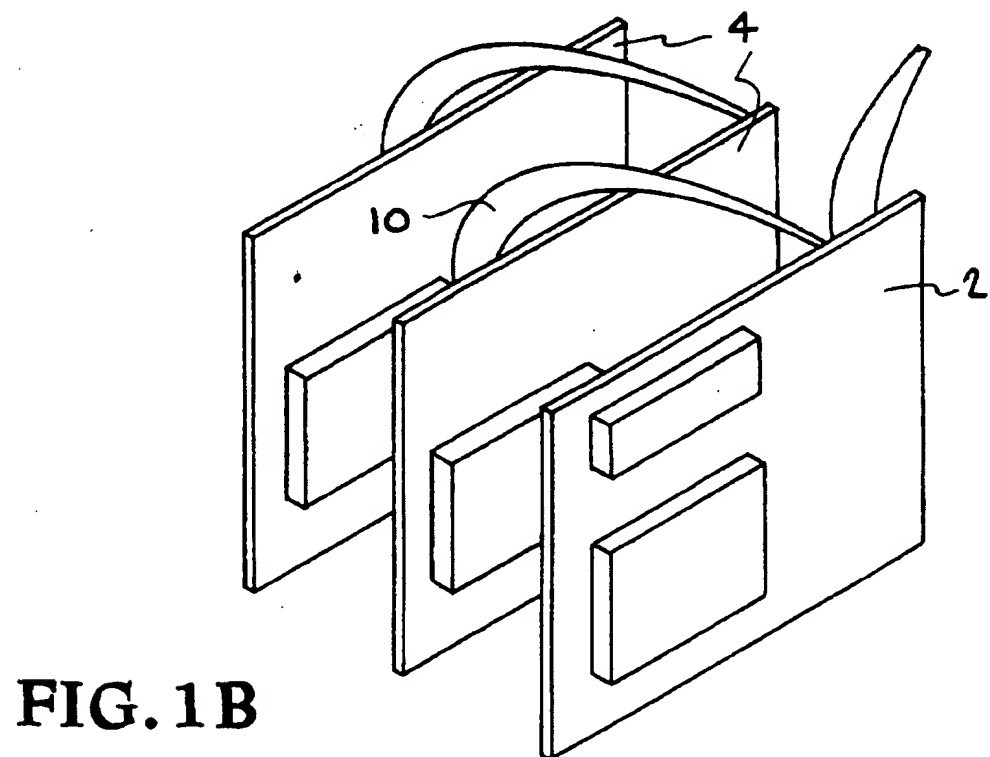
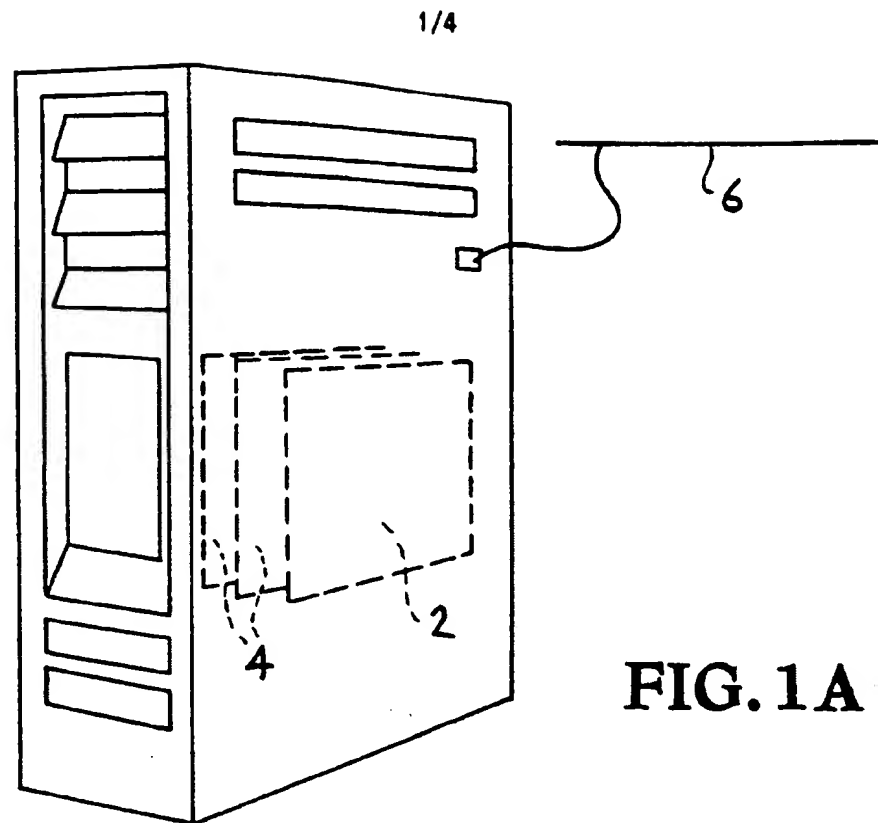
15 a multi-threaded, symmetric-multi-processor capable operating system running on said plurality of CPUs and said second plurality of CPUs, wherein said operating system provides services and coordinates concurrent execution of said parallel calculations.

4. The engine of claim 3, wherein said parallel dose calculations comprise a computer implemented process for producing a 3-dimensional map of a radiation dose delivered to a patient, comprising:

5 constructing patient-dependent information necessary for a Monte-Carlo transport calculation;

executing said Monte-Carlo transport calculation; and

10 producing, from said patient-dependent information and said Monte-Carlo transport calculation, a 3-dimensional map of the dose delivered to said patient.



2/4

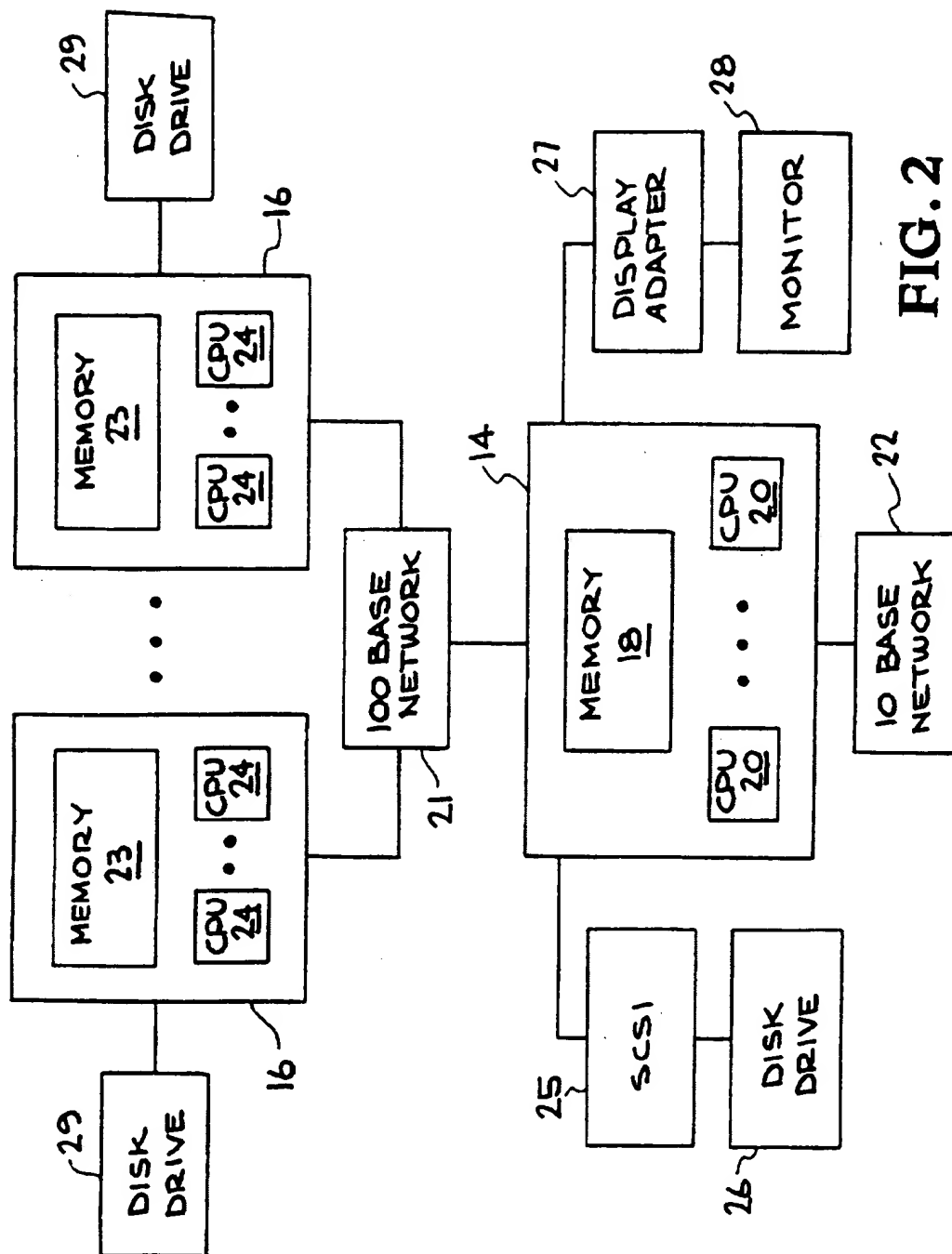
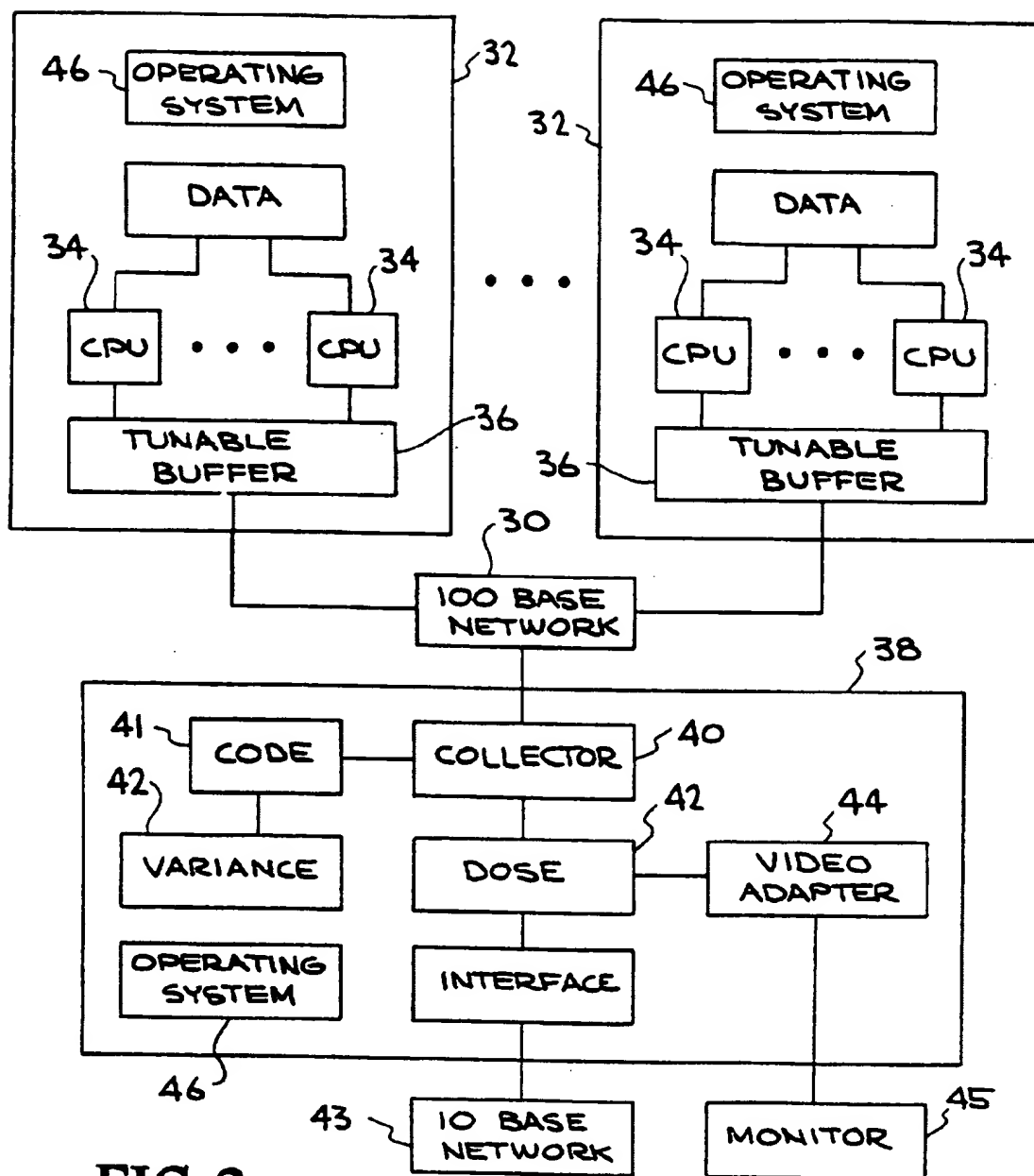


FIG. 2

3/4

**FIG. 3**

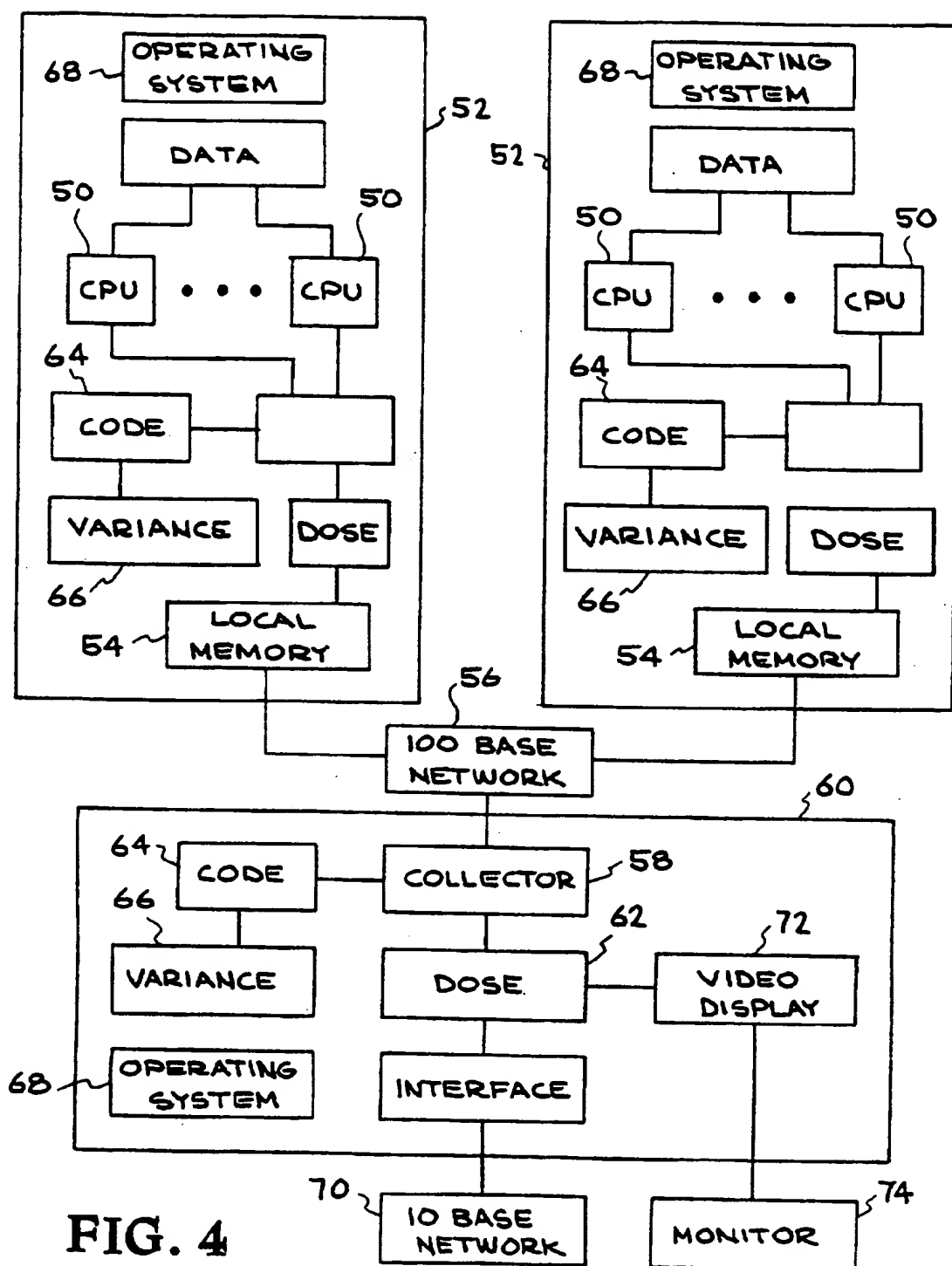


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/07769

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G01T 1/161

US CL : 395/800; 128/920

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/800; 128/920; 653.1; 364/554, 555

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, IEEE, MAYA, Computer Select

search terms: radiation dosage monte carlo smp multiprocessor motherboard

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,205,289 (HARDY ET AL) 27 April 1993, see entire document, especially abstract, col. 3, lines 35-61, and col. 5, lines 10-35	1-4
Y	RODENAS et al. 'Application of the Monte Carlo Method to Estimate the Tenth-value Thickness for X-rays in Medical Electron Accelerators' In: Proceedings of the 1993 Particle Accelerator Conference. New York, NY. IEEE, 1993. p. 647-649	1-4
Y	MARR et al. 'Multiprocessor Validation of the Pentium Pro Microprocessor' In: COMPCON '96 Technologies for the Information Superhighway. Los Alamitos, CA. IEEE Comput. Soc. Press, 1996. p. 395-400	1-4

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be part of particular relevance	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Z*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

22 JULY 1997

Date of mailing of the international search report

29 AUG 1997

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Authorized officer

Christopher S. Chow

Facsimile No. (703) 305-3230

Telephone No. (703) 308-6674